HT32 低功耗模式

文件编码：AN0473S

概述

简介

此篇应用范例用于帮助系统设计者对 HT32 系列产品的低功耗模式，在软硬件方面有简要地了解。用户可以学习使用 HT32 产品及如何安排低功耗模式，以实现在低功耗应用方面的最优化处理。

相关档

- HT32 系列规格书
- HT32 系列使用手册

功耗和唤醒时间测量

简介

阅读此节后，用户可以了解 HT32 系列低功耗模式功耗和唤醒时间的测量问题。

功耗测量

用户可通过此应用范例附带的 Zip 文件中提供的韧体实际测量 HT32F52352 的功耗。该韧体存放在 CurrentMeasurements 文件夹中。

可以在下列的低功耗模式中进行功耗测量：

- 休眠模式
  休眠模式下仅有 CPU 内核的时钟停止运行。此模式下的功耗可由时钟源及有效的外设决定。为涵盖此模式的所有功能，测量时用到不同时钟源 (HSI 和 HSE)、时钟频率 (1MHz~48MHz)、APB 外设配置 (所有外设时钟开启或所有外设时钟关闭，包括 Flash 和 SRAM)。
- 深度休眠模式 1
  深度休眠模式 1 中，1.5V 电源域的所有时钟停止运行。高速振荡器 (HSI、HSE) 和 PLL 停止振荡，1.5V 电压源为 LDO (处于低电流模式时)。
- 深度休眠模式 2
  除了 1.5V 电压源为 DMOS 外，深度休眠模式 2，其余条件与深度休眠模式 1 相同。

- 暂停模式

  关闭 1.5V 电源域，保留 V_{DD} 和 Backup 电源域。

注意：1. 更多不同低功耗模式的说明，请参考 HT32 系列的用户手册及规格书。

2. 处于休眠模式和深度休眠模式时，所有未使用的 I/O 脚都配置为输入浮空状态，斯密特触发输入除能，因此，这些 I/O 引脚的功耗为 0。

如何测量功耗

HT32 系列开发板的功耗测量可通过两种方式测得：首先利用电表代替 MCU Power Jumper (以 HT32F52352 Starter Kit 为例 Power Jumper 是 J1)，并且由外部供电给开发板；或者直接使用 USB 线供电。

韧体描述

头文件(main.h)中，可通过选择几个 #define 参数化范例。

- 取消相应行定义的注释，选择休眠模式所需的时钟配置

```
#define HSE_PLL_ON
#define HSE_PLL_ON_48MHz
//#define HSE_PLL_ON_24MHz
//#define HSE_PLL_ON_12MHz
//#define HSE_PLL_OFF
//#define HSE_PLL_OFF_8MHz
//#define HSE_PLL_OFF_1MHz
//#define HSI_PLL_ON
//#define HSI_PLL_ON_48MHz
//#define HSI_PLL_ON_24MHz
//#define HSI_PLL_ON_12MHz
//#define HSI_PLL_OFF
//#define HSI_PLL_OFF_8MHz
//#define HSI_PLL_OFF_1MHz
```

- 取消相应行定义的注释，选择所需的低功耗模式

```
#define SLEEP
//#define DEEP_SLEEP_1
//#define DEEP_SLEEP_2
//#define POWER_DOWN
```

- 取消相应行定义的注释，选择所需外设时钟 (门控时钟)

```
//#define SLEEP_ALLPERIPH_ENABLE
#define SLEEP_ALLPERIPH_DISABLE
```

- 休眠模式下，用户可通过取消相应行定义的注释来选择所需的 FMC 时钟选项

```
//#define SLEEP_FMC_ENABLE
#define SLEEP_FMC_DISABLE
```

- 休眠模式下，用户可通过取消相应行定义的注释来选择所需的 SRAM 时钟选项

```
//#define SLEEP_SRAM_ENABLE
//#define SLEEP_SRAM_DISABLE
```

注意：无论时钟源来自 HSI 还是 HSE，若系统时钟频率等于或小于 8MHz，PLL 都将关闭。执行低功耗范例后，若用户想重新载入 Flash 内存的内容，需先将启动模式由主 Flash 切换到 Boot Loader (Boot Option 在 Starter Kit 板子背面)，并且要按下复位键。此举是因为当 HT32 系列处于低功耗模式时，调试器无法连接到 HT32。接着启动模式需重新配置到主 Flash。而后，重启目标板开始进行测量。

测量结果

测量结果如表 1 和表 2 所示。(关于低功耗数据，可参考 HT32 系列规格书)

条件	f_{HCLK}	所有 APB 外设使能	所有 APB 外设除能
运行 HSE。 AHB 预分频器用于减小频率， 休眠模式下 FMC 和 SRAM 时钟开启	48MHz	9.92mA	2.85mA
	24MHz	4.92mA	1.94mA
	12MHz	3.47mA	1.47mA
	8MHz	1.89mA	0.85mA
	1MHz	0.76mA	0.57mA
运行 HSE。 AHB 预分频器用于减小频率， 休眠模式下 FMC 和 SRAM 时钟关闭	48MHz	9.44mA	2.38mA
	24MHz	4.77mA	1.69mA
	12MHz	3.36mA	1.34mA
	8MHz	1.81mA	0.76mA
	1MHz	0.69mA	0.56mA
运行 HSI。 AHB 预分频器用于减小频率， 休眠模式下 FMC 和 SRAM 时钟关闭	48MHz	9.77mA	2.65mA
	24MHz	4.95mA	1.96mA
	12MHz	3.61mA	1.61mA
	8MHz	2.09mA	1.04mA
	1MHz	0.97mA	0.84mA

表 1　休眠模式下功耗测量结果(以 HT32F52352 为范例)

模式	条件	V_{DD}/V_{BAT} = 3.3V
深度休眠 1 模式	VDD15 来自 LDO，LDO 处于低功耗模式，RTC on，使用 LSI	42.5 μA
深度休眠 2 模式	VDD15 来自 DMOS，LDO Off，RTC on，使用 LSI	13.3 μA
暂停模式	VDD15 Off，RTC on，使用 LSI	1.5 μA

表 2　深度休眠模式和暂停模式下功耗测量(以 HT32F52352 为范例)

唤醒时间测量

此节描述了如何测量 HT32 系列由不同低功耗模式下唤醒的时间。此应用范例提供了相关的韧体，韧体位于提供的 Zip 压缩包内的 Wakeup Timing 以及 Wakeup Timing(PowerDownMode) 文件夹下。

唤醒时间定义

- 休眠模式和深度休眠模式

  唤醒时间起始于 RTCOUT (以 HT32F52352 为例脚位是 PB12) 的上升沿，结束于 WFE 后执行完第一条指令。

- 暂停模式

  从暂停模式中唤醒后，与遇到复位条件相同，程序重新执行。暂停模式的唤醒时间介于 RTCOUT (以 HT32F52352 为例脚位是 PB12) 的上升沿与执行完第一条指令之间。

韧体描述

头文件(main.h)中，可通过选择几个 #define 参数化范例。

- 取消相应行定义的注释，选择所需的低功耗模式

```
//#define SLEEP
//#define DEEP_SLEEP_1
//#define DEEP_SLEEP_2
```

- 取消相应行的注释，选择所需的系统时钟源

```
/* Define the system clock */
#define HCLK_HSI
//#define HCLK_HSI_PLL
//#define HCLK_HSE
//#define HCLK_HSE_PLL
```

在主文档(main.c)中。进入低功耗模式前，PA5 配置为输出推挽式，复位到低电压状态。由低功耗模式唤醒后：

- 如果从休眠模式和深度休眠模式唤醒，则执行直接写入 GPIOA_SRR (输出设定复位控制寄存器) 以设定 PA5 为高电平。
- 如果从暂停模式唤醒，则在代码启动时 PA5 脚应被设定 (此部分代码以 HT32F52352 为例在 startup_ht32f520xx_01.s，如下所示)。

```
;/* Enable peripheral clocks of AFIO and GPIOA */
LDR R0, = 0x4008802C
LDR R1, = 0x00004000
STR R1,[R0]

LDR R0, = 0x40088024
LDR R1, = 0x00010000
STR R1,[R0]
;/* PA5 output high */
LDR R0, = 0x4001A000
LDR R1, = 0x0020
STR R1,[R0, #0x24]
STR R1,[R0]
```

注意：执行低功耗范例后，若用户想重载 Flash 内存的内容，需将启动模式由主 Flash 切换到 Boot Loader，并且要按下复位键。此举是因为当 HT32 系列处于低功耗模式时，调试器无法连接到 HT32。启动模式需重新配置到主 Flash。而后，通过重启目标板开始进行测量。

测量结果

休眠模式、深度休眠模式和暂停模式的唤醒时间测量结果如表 3 所示。

符号	参数	条件	典型值
$t_{WUSLEEP}$	从休眠模式中唤醒	在 HSI 时钟下唤醒	1.00 μs
t_{WUDS1}	从深度休眠模式 1 中唤醒		11.00 μs
t_{WUDS2}	从深度休眠模式 2 中唤醒		11.0 μs
t_{WUPD}	从暂停模式中唤醒		110 μs
$t_{WUSLEEP}$	从休眠模式中唤醒	在 HSE 时钟下唤醒	1.90 μs
t_{WUDS1}	从深度休眠模式 1 中唤醒		4.30ms
t_{WUDS2}	从深度休眠模式 2 中唤醒		4.33ms
$t_{WUSLEEP}$	从休眠模式中唤醒	在 PLL 时钟下唤醒 PLL 使用 HSI	0.30 μs
t_{WUDS1}	从深度休眠模式 1 中唤醒		136 μs
t_{WUDS2}	从深度休眠模式 2 中唤醒		136 μs
$t_{WUSLEEP}$	从休眠模式中唤醒	在 PLL 时钟下唤醒 PLL 使用 HSE	0.30 μs
t_{WUDS1}	从深度休眠模式 1 中唤醒		4.43ms
t_{WUDS2}	从深度休眠模式 2 中唤醒		4.44ms

表 3　唤醒时间测量结果 (以 HT32F52352 为范例)

结论

根据不同的结果，用户可以在功耗和唤醒时间之间作一权衡，前提条件是：功耗更低、唤醒时间更长。

由于 HSE 和 PLL 需更长的准备时间，故当时钟源为 HSE 或 PLL 时，深度休眠模式唤醒时间更长。为缩短深度休眠模式的唤醒时间，用户可以先使用 HSI，然后切换到其他外设时钟源 (系统时钟、HSE 或 PLL)。

根据应用的限制，用户可选择最好的权衡结果。

功耗优化

简介

事实上：单片机的功耗随着时钟频率的增大而增加。所以，用户须寻找到最优的功耗/性能比。很多应用中，可通过调整系统/外设频率达到所需的性能而降低功耗。若系统/外设工作无特别需求，可使用 HT32 的低功耗模式。

应用中使用时钟配置

本节介绍如何使用 HT32 的时钟配置，所使用的韧体位于本应用范例的 Zip 压缩包内的 RunMode 文件夹中。

该程序使用 USART 从 RTC 传送时间。

- 首先，用户必须使用超级终端调整时间
- 时间显示在超级终端上，并且每秒刷新一次。RTC 设定为每秒产生一次中断
- 当中断发生时，捕捉 RTC 计数器的值；计算时间，并使用 USART 传送出去

硬件环境

在 HT32F52352 Starter Kit 上使用此范例：请参考 HT32F5xxxx Starter Kit 的用户手册，以及 e-Link32/e-Link32 Pro 用户手册。

- 使用 Starter Kit 上的 Serial-Wire Debugger，将开发板与 PC 连接
- 由一个电表代替跳线 J1 测量功耗

韧体描述

- 在 PC 上配置超级终端
  - 字长 = 8 bits
  - 一个 Stop 位
  - 无奇偶校验
  - 波特率 = 115200
  - 流程控制：无

- 配置韧体

  头文件(main.h)中，可通过选择几个 #define 参数化范例。

  - 定义外设选择 (门控时钟)

```
//#define ALL_PERIPHERIALS_ENABLE
//#define ONLY_USART_RTC_ENABLE □
```

  - 定义频率选择

```
//#define HCLK_48MHz
//#define HCLK_8MHz □
```

  - 定义应用程序等待 RTC 中断时切换到休眠模式

```
//#define SLEEP_WFI_ON
```

注意：执行低功耗范例后，若用户想重载 Flash 内存的内容，需将启动模式由主 Flash 切换到 Boot Loader，并且要按下复位键。此举是因为当 HT32 处于低功耗模式时，调试器无法连接到 HT32。启动模式需重新配置到主 Flash。而后，通过重启目标板开始进行测量。

测量结果

测量结果如表 4 所示。

外设时钟	频率	休眠	典型功耗
ALL On	48MHz	No	20.33mA
仅 RTC On	48MHz	No	11.80mA
仅 RTC On	48MHz	Yes	4.35mA
仅 RTC On	8MHz	No	2.92mA
仅 RTC On	8MHz	Yes	1.66mA

表 4　在 25℃ 时运行模式测量举例 (以 HT32F52352 为范例)

结论

为减少功耗，必须根据用户需求以最优化配置进行初始化。用户必须着眼于应用的要求，并配置相应的 HT32 功能。从这个范例中，用户可以借鉴可能的 HT32 时钟配置和优化应用程序的功耗。

配置描述如下：

- 系统和外设频率

  若应用程序无需在最大频率下运行，用户可通过使用 PLL 或预分频器除频以减小 HCLK。

- 门控时钟

  为优化功耗，用户应通过 HT32 门控时钟选项，除能未使用的外设。

- 休眠模式

  减少功耗的另一个方法，是当应用程序等待事件或者中断时，切换到 HT32 的休眠模式。

在电池产品应用中使用深度休眠模式和暂停模式

简介

由电池供电的一些应用程序不是一直处于运行中的。此种应用中，单片机等待外部事件，且需要在暂停运行期间减少功耗。

此应用范例提供了两个，在电池供电应用中如何使用 HT32 的范例 (深度休眠模式和暂停模式)。这些范例中，一旦应用程序无需处理，HT32 就切换到低功耗模式。这两个低功耗模式基于 Cortex® - M0+/M3 内核的深度睡眠功能和 WFE 指令。

深度休眠模式

本节介绍如何使用 HT32 的 WFE 指令和深度休眠模式，所使用的韧体位于本应用范例的 Zip 压缩包内的 DeepSleepMode 文件夹中。

该范例执行定期 ADC 转换并将转换结果记忆在 RAM 缓冲中。它采用 RTC 比较匹配事件以自动唤醒深度休眠模式。

- 首先，配置备份域、GPIO 和 ADC
- RTC 正在运行，且由主循环中的 RTC 比较匹配触发每次 ADC 转换和 ADC 转换结果的记忆
- RTC 比较匹配事件将 HT32 由深度休眠模式中唤醒
- PA5 I/O 显示了 ADC 转换和 RTC_CMP 寄存器重载所需的时间

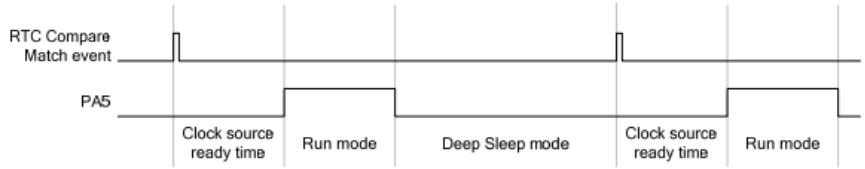


图 1　深度休眠模式举例

韧体描述

头文件(main.h)中，可通过选择几个 #define 参数化范例。

- 取消相应模式定义的注释，选择所需的低功耗模式

```
//#define DEEP_SLEEP_1
//#define DEEP_SLEEP_2
```

- 选择循环时序，取消相应行的注释

```
//#define LOOP_50ms
//#define LOOP_500ms
//#define LOOP_1s
//#define LOOP_3s
//#define LOOP_5s
```

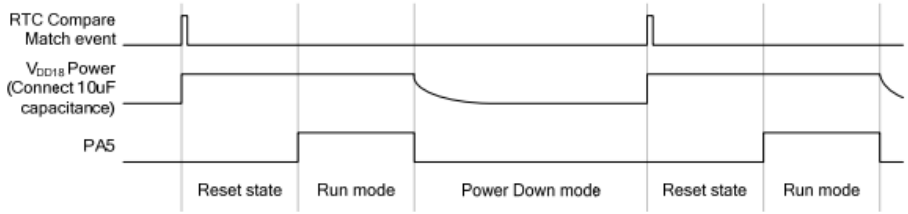
注意：执行低功耗范例后，若用户想重载 Flash 内存的内容，需将启动模式由主 Flash 切换到 Boot Loader，并且要按下复位键。此举是因为当 HT32 处于低功耗模式时，调试器无法连接到 HT32。启动模式需重新配置到主 Flash。而后，通过重启目标板开始进行测量。

暂停模式

本节介绍如何使用 HT32 的暂停模式，所使用的韧体位于本应用范例的 Zip 压缩包内的 PowerDownMode 文件夹中。

该范例执行定期 ADC 转换并将转换结果记忆在 RAM 缓冲中。它采用 RTC 比较匹配事件以自动唤醒暂停模式。

- 首先，配置备份域、GPIO 和 ADC
- RTC 运行且 RTC 比较匹配事件将 HT32F523xx 由暂停模式中唤醒
- 每次由暂停模式中唤醒后，HT32F523xx 从复位状态重启，且在 HT32F523xx 已配置后执行每次 ADC 转换
- PA5 I/O 显示了程序运行所需的时间



韧体描述

头文件(main.h)中，可通过选择几个 #define 参数化范例。

- 选择循环时序，取消相应行的注释

```
//#define LOOP_50ms
//#define LOOP_500ms
//#define LOOP_1s
//#define LOOP_3s
//#define LOOP_5s
```

注意：执行低功耗范例后，若用户想重载 Flash 内存的内容，需将启动模式由主 Flash 切换到 Boot Loader，并且要按下复位键。此举是因为当 HT32 处于低功耗模式时，调试器无法连接到 HT32。启动模式需重新配置到主 Flash。而后，通过重启目标板开始进行测量。

如何测量电流消耗

以 HT32F52352 Starter kit 为例，功耗测量可利用电表代替跳线 J1，并且通过使用 USB 线供电给开发板。

测量结果

测量结果如表 5 所示。

低功耗模式	50ms	500ms	1s	3s	5s
深度休眠模式 1	46 μA	44.7 μA	44.7 μA	44.6 μA	44.5 μA
深度休眠模式 2	20.6 μA	17.6 μA	16.3 μA	16.1 μA	16.0 μA
暂停模式	280 μA	36.1 μA	25.8 μA	15.1 μA	11.7 μA

表 5　在 25°C 下低功耗模式测量结果举例 (以 HT32F52352 为范例)

结论

由于在 Cortex®-M0+/M3 内核集成了用于低功耗应用 (休眠模式和深度休眠模式) 的高效核心级指令，同时结合 HT32 的低功耗特性，使用户可以根据应用的需求来优化系统功耗。

测量表明，必须考虑唤醒时间和功耗之间的最优化。

版本及修改信息

Date 日期	Author 作者	Issue 发行、修订说明
2017.11.20	吴旭宏	第一版

免责声明

本网页所载的所有数据、商标、图片、链接及其他数据等 (以下简称「数据」)，只供参考之用，盛群半导体股份有限公司 (以下简称「本公司」) 将会随时更改数据，并由本公司决定而不作另行通知。虽然本公司已尽力确保本网页的数据准确性，但本公司并不保证该等数据均为准确无误。本公司不会对任何错误或遗漏承担责任。

本公司不会对任何人士使用本网页而引致任何损害 (包括但不限于计算机病毒、系统固障、数据损失) 承担任何赔偿。本网页可能会连结至其他机构所提供的网页，但这些网页并不是由本公司所控制。本公司不对这些网页所显示的内容作出任何保证或承担任何责任。

责任限制

在任何情况下，本公司并不须就任何人由于直接或间接进入或使用本网站，并就此内容上或任何产品、信息或服务，而招致的任何损失或损害负任何责任。

管辖法律

本免责声明受中华民国法律约束，并接受中华民国法院的管辖。

免责声明更新

本公司保留随时更新本免责声明的权利，任何更改于本网站发布时，立即生效。